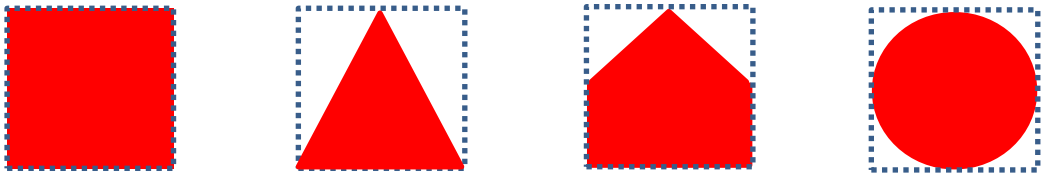
Δεύτερη Άσκηση
Ημερομηνία Παράδοσης: 4 Ιουνίου, 2021, 3 μ.μ.

Για την άσκηση αυτή θα υλοποιήσετε ένα απλό παιχνίδι με σχήματα που μοιάζει (λίγο) με το Tetris. Έχουμε μια γεννήτρια που παράγει σχήματα επιλεγμένα τυχαία από τετράγωνο, τρίγωνο, πεντάγωνο και κύκλο (όπως φαίνονται παρακάτω). Ο συνολικός αριθμός των σχημάτων που μπορεί να παράγει η γεννήτρια είναι προκαθορισμένος, και τα σχήματα παράγονται με τυχαία επιλεγμένο μέγεθος. Έχουμε ένα παίχτη ο οποίος μαζεύει τα σχήματα σε μία στοίβα περιορισμένου μεγέθους. Για κάθε νέο σχήμα που παράγεται, ο παίχτης πρέπει να επιλέξει αν θα κρατήσει αυτό το σχήμα ή όχι. Αν το κρατήσει, το σχήμα μπαίνει στην κορυφή της στοίβας. Για κάθε σχήμα που κρατάει, ο παίχτης παίρνει ένα αριθμό πόντων ίσο με το εμβαδόν του σχήματος. Αν το καινούριο σχήμα που μαζέψει ο παίχτης έχει το ίδιο εμβαδόν με το αυτό στην κορυφή της στοίβας τότε οι πόντοι που παίρνει δεκαπλασιάζονται. Επίσης, αν το νέο σχήμα είναι ίδιου τύπου με αυτό στην κορυφή της στοίβας τότε τα δύο αυτά σχήματα αφαιρούνται από την στοίβα (ο παίχτης κρατάει τους πόντους). Το παιχνίδι τελειώνει όταν είτε τελειώσουν τα σχήματα της γεννήτριας, είτε γεμίσει η στοίβα του παίχτη. Ο στόχος του παίχτη είναι να συγκεντρώσει όσο περισσότερους πόντους γίνεται.



Για την υλοποίηση σας θα δημιουργήσετε τις παρακάτω κλάσεις:

1. Την **αφηρημένη** κλάση **Shape** η οποία κρατάει πληροφορίες για ένα γενικό σχήμα. Έχει ένα μόνο πεδίο, ένα int με το εμβαδόν του τετραγώνου που περικλείει το σχήμα (περιβάλλον τετράγωνο – bounding rectangle). Το περιβάλλον τετράγωνο είναι αυτό που φαίνεται με διακεκομμένες γραμμές παραπάνω. Το πεδίο αρχικοποιείται στον constructor. Η κλάση έχει τις εξής μεθόδους:
   - Την **αφηρημένη** μέθοδο **computeArea** η οποία επιστρέφει το εμβαδόν του σχήματος (double).
   - Την **αφηρημένη** μέθοδο **getType** η οποία επιστρέφει ένα String με τον τύπο του σχήματος.
   - Την ενυπόστατη (concrete) μέθοδο **sameArea** η οποία παίρνει σαν όρισμα ένα άλλο σχήμα και συγκρίνει αν έχουν το ίδιο εμβαδόν.
   - Την ενυπόστατη (concrete) μέθοδο **sameType** η οποία παίρνει σαν όρισμα ένα άλλο σχήμα και συγκρίνει αν έχουν τον ίδιο τύπο.
   - Την ενυπόστατη (concrete) μέθοδο **toString** η οποία επιστρέφει ένα String με τον τύπο και εμβαδόν του σχήματος.
   - Μέθοδο πρόσβασης για το εμβαδόν του περιβάλλοντος τετραγώνου.

Αποθηκεύστε τον κώδικα σας στο αρχείο **Shape.java**, το οποίο θα παραδώσετε.

2. Τέσσερεις ενυπόστατες κλάσεις **Square**, **Triangle**, **Pentagon**, **Circle** οι οποίες κληρονομούν από την κλάση Shape. Ο constructor παίρνει σαν όρισμα το εμβαδόν του περιβάλλοντος τετραγώνου. Οι κλάσεις υλοποιούν τις αφηρημένες μεθόδους. Αν B είναι το εμβαδόν του περιβάλλοντος τετραγώνου, το εμβαδόν του τετραγώνου είναι (επίσης) B, του τριγώνου ½ B, του πενταγώνου ¾ B και του κύκλου $\frac{\pi}{4}$B. Το String με τον τύπο του σχήματος θα είναι ίδιο με το όνομα της κλάσης.

Υπόδειξη: Χρησιμοποιήστε το Math.PI για την τιμή του π.

Αποθηκεύστε τον κώδικα σας στα αντίστοιχα **.java** αρχεία, τα οποία θα παραδώσετε.

3. Την κλάση **ShapeGenerator** η οποία υλοποιεί την γεννήτρια που παράγει σχήματα. Θα δημιουργήσετε ίδιο αριθμό από σχήματα για κάθε τύπο. Ο constructor της ShapeGenerator παίρνει σαν όρισμα τον αριθμό των σχημάτων ανά τύπο και δημιουργεί έναν πίνακα με όλα τα σχήματα σε τυχαία διάταξη. Εκτός από τον πίνακα με τα σχήματα, ορίστε όποια επιπλέον πεδία χρειάζεστε για την υλοποίηση των μεθόδων της κλάσης.
Η κλάση έχει τις εξής μεθόδους:

- Μία private μέθοδο **initializeShapes** που καλείται από τον constructor και γεμίζει τον πίνακα με τα σχήματα. Δημιουργούμε τον ίδιο αριθμό από σχήματα για κάθε τύπο. Το εμβαδόν του περιβάλλοντος τετραγώνου για κάθε σχήμα επιλέγεται τυχαία από ένα πίνακα με τα πιθανά μεγέθη. Για την υλοποίηση σας βάλετε τις τιμές {1,2,3,4,8,12,16} στον πίνακα.
- Την μέθοδο **nextShape** η οποία επιστρέφει ένα τυχαίο σχήμα από τον πίνακα και το αφαιρεί.
- Την μέθοδο **hasShape** που επιστρέφει μια Boolean τιμή αν η γεννήτρια έχει άλλα σχήματα.
- Την μέθοδο **toString** που τυπώνει τα σχήματα που έχει στον πίνακα η γεννήτρια. Την μέθοδο αυτή πρέπει να την υλοποιήσετε και να την παραδώσετε, αλλά θα την χρησιμοποιήσετε μόνο για debugging.
- Μια μέθοδο **main** που δημιουργεί μια γεννήτρια με 2 σχήματα ανά τύπο, και καλεί την nextShape μέχρι να τελειώσουν τα σχήματα της γεννήτριας. Σε κάθε επανάληψη τυπώνει το σχήμα που επέστρεψε και τα περιεχόμενα της γεννήτριας. Την μέθοδο αυτή πρέπει να την υλοποιήσετε και να την παραδώσετε, αλλά θα την χρησιμοποιήσετε μόνο για debugging.

Υποδείξεις: Η προτεινόμενη υλοποίηση για τη γεννήτρια είναι να χρησιμοποιήσετε πίνακα για να κρατάτε τα αντικείμενα. Στην περίπτωση αυτή τα σχήματα θα πρέπει να είναι διατεταγμένα σε τυχαία σειρά. Μπορείτε να χρησιμοποιήσετε παρόμοια υλοποίηση όπως την πρώτη άσκηση για να πάρετε μια τυχαία αναδιάταξη των σχημάτων στην initializeShapes. Η κλήση της nextShape δεν χρειάζεται να αφαιρεί το σχήμα από τον πίνακα, αρκεί να μην το «βλέπουμε».
Εναλλακτικά, μπορείτε να χρησιμοποιήσετε οποιαδήποτε άλλη δομή θέλετε για να αποθηκεύσετε τα σχήματα και να υλοποιήσετε την nextShape αντίστοιχα. Όποια και αν είναι η υλοποίηση σας, θα πρέπει να έχετε μόνο μία δομή (π.χ., πίνακα) με σχήματα, δεν μπορείτε να έχετε διαφορετικές δομές για κάθε τύπο.

Αποθηκεύστε τον κώδικα σας στο αρχείο **ShapeGenerator.java**, το οποίο θα παραδώσετε.

4. Την κλάση **Player** η οποία υλοποιεί τον παίχτη. Η κλάση θα κρατάει οπωσδήποτε την στοίβα με τα σχήματα που μαζεύει ο παίχτης, και τους πόντους του παίχτη. Ο constructor παίρνει σαν όρισμα το μέγεθος της στοίβας. Μπορείτε να υλοποιήσετε την στοίβα όπως θέλετε (π.χ., με πίνακα όπως κάναμε στο μάθημα, ή χρησιμοποιώντας κάποια υπάρχουσα βιβλιοθήκη της Java όπως την Stack, ArrayDeque, ή LinkedList). Ορίστε και όποιο άλλο πεδίο χρειάζεστε.
Η κλάση θα έχει τις εξής μεθόδους:

- Την μέθοδο **playShape** η οποία παίρνει σαν όρισμα ένα σχήμα και υλοποιεί το παιχνίδι του παίχτη για το σχήμα. Θα πρέπει να ρωτάει τον παίχτη αν θέλει να το κρατήσει ή όχι και να ενημερώνει τα πεδία κατάλληλα. Η μέθοδος υποθέτει ότι υπάρχει χώρος στην στοίβα για να αποθηκευτεί το νέο σχήμα. Μπορείτε προαιρετικά να εκτυπώνετε επιπλέον πληροφορίες (π.χ. τους πόντους που πήρε ο παίχτης, ή αν αφαιρέθηκαν τα σχήματα από την κορυφή της στοίβας).
- Την μέθοδο **isStackFull** που ελέγχει αν γέμισε η στοίβα του παίχτη.
- Την μέθοδο **printStack** που εκτυπώνει την στοίβα. Παράδειγμα της εκτύπωσης σας δίνεται παρακάτω.
- Μέθοδο πρόσβασης για τους πόντους του παίχτη.

Αποθηκεύστε τον κώδικα σας στο αρχείο **Player.java**, το οποίο θα παραδώσετε.

5. Την κλάση **ShapeGame** η οποία θα έχει την main που υλοποιεί το παιχνίδι. Θα ζητήσετε από την είσοδο τον αριθμό των σχημάτων ανά τύπο, και το μέγεθος της στοίβας του παίχτη και θα δημιουργήσετε την γεννήτρια και τον παίχτη. Όσο η γεννήτρια έχει σχήματα και η στοίβα του παίχτη δεν έχει γεμίσει συνεχίζεται το παιχνίδι. Πριν από κάθε σχήμα τυπώστε την στοίβα του παίχτη, και μετά από κάθε σχήμα τυπώστε τους πόντους του. Παραδείγματα εξόδου σας δίνονται παρακάτω.

Αποθηκεύστε τον κώδικα σας στο αρχείο **ShapeGame.java**, το οποίο θα παραδώσετε.

ΓΕΝΙΚΕΣ ΟΔΗΓΙΕΣ

- Μια κλάση που δεν κάνει compile μηδενίζεται αυτόματα.
- Δεν επιτρέπεται η χρήση public ή protected πεδίων στην άσκηση. Επίσης ο κώδικας θα πρέπει να είναι σωστά στοιχισμένος και καλά γραμμένος. Θα αφαιρεθούν βαθμοί από προγράμματα που είναι πολύ κακά γραμμένα.
- Θα τεστάρουμε και θα βαθμολογήσουμε την κάθε κλάση ξεχωριστά. Γι αυτό και θα πρέπει να σώσετε την κάθε κλάση σε ξεχωριστό αρχείο. Θα πρέπει επίσης να κρατήσετε τα ονόματα και τα ορίσματα των public μεθόδων ακριβώς όπως σας ζητούνται.
- Κάντε turnin τα προγράμματα σας στο assignment2@myy205.

  π.χ. turnin assignment1@myy205 ShapeGame.java

  Μπορείτε να κάνετε turnin πολλά αρχεία μαζί στην ίδια εντολή (π.χ. turnin assignment1@myy205 *.java παραδίδει όλα τα java αρχεία στο folder). Διαβάστε προσεκτικά τις οδηγίες για το trunin στο ecourse και βεβαιωθείτε ότι μπορείτε να κάνετε την διαδικασία κάποιες μέρες πριν την προθεσμία. Κάθε φορά πρέπει να κάνετε turnin όλα τα αρχεία που θέλετε να παραδώσετε. Μπορείτε να κάνετε πολλαπλές φορές turnin τα ίδια αρχεία, θα κοιτάξουμε το τελευταίο. Δεν μπορείτε να κάνετε turnin zip αρχείο, ή αρχείο με ελληνικούς χαρακτήρες.

  Στον κώδικα να αναγράφονται σε σχόλια το όνομα και ο ΑΜ σας (**με λατινικούς χαρακτήρες**).

Παράδειγμα Εξόδου:

Παρακάτω δίνεται ένα ενδεικτικό παράδειγμα της εξόδου του παιχνιδιου. Δεν είναι ανάγκη η έξοδος σας να είναι ακριβώς έτσι αλλά πρέπει να είναι παρόμοια.

```
assignment2>java ShapeGame
Give the number of shapes per type
25
Give the size of the stack for the player
10

Incoming shape Triangle:0.5
Do you want to accept? (y/n)
y
Added 0.5 points
Player has 0.5 points

Current stack:
        1:Triangle:0.5

Incoming shape Triangle:2.0
Do you want to accept? (y/n)
y
Added 2.0 points
Top shapes removed
Player has 2.5 points

Current stack:

Incoming shape Circle:9.42477796076938
Do you want to accept? (y/n)
y
Added 9.42477796076938 points
Player has 11.92477796076938 points

Current stack:
        1:Circle:9.42477796076938
```

```
Incoming shape Pentagon:0.75
Do you want to accept? (y/n)
y
Added 0.75 points
Player has 12.67477796076938 points

Current stack:
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Circle:0.7853981633974483
Do you want to accept? (y/n)
y
Added 0.7853981633974483 points
Player has 13.460176124166829 points

Current stack:
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Triangle:8.0
Do you want to accept? (y/n)
y
Added 8.0 points
Player has 21.46017612416683 points

Current stack:
        4:Triangle:8.0
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Pentagon:1.5
Do you want to accept? (y/n)
n
Player has 21.46017612416683 points

Current stack:
        4:Triangle:8.0
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Square:8.0
Do you want to accept? (y/n)
y
10X points!
Added 80.0 points
Player has 101.46017612416682 points

Current stack:
        5:Square:8.0
        4:Triangle:8.0
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Triangle:2.0
Do you want to accept? (y/n)
y
Added 2.0 points
Player has 103.46017612416682 points

Current stack:
        6:Triangle:2.0
```

```
        5:Square:8.0
        4:Triangle:8.0
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Square:8.0
Do you want to accept? (y/n)
y
Added 8.0 points
Player has 111.46017612416682 points

Current stack:
        7:Square:8.0
        6:Triangle:2.0
        5:Square:8.0
        4:Triangle:8.0
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Triangle:6.0
Do you want to accept? (y/n)
y
Added 6.0 points
Player has 117.46017612416682 points

Current stack:
        8:Triangle:6.0
        7:Square:8.0
        6:Triangle:2.0
        5:Square:8.0
        4:Triangle:8.0
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Square:3.0
Do you want to accept? (y/n)
y
Added 3.0 points
Player has 120.46017612416682 points

Current stack:
        9:Square:3.0
        8:Triangle:6.0
        7:Square:8.0
        6:Triangle:2.0
        5:Square:8.0
        4:Triangle:8.0
        3:Circle:0.7853981633974483
        2:Pentagon:0.75
        1:Circle:9.42477796076938

Incoming shape Pentagon:9.0
Do you want to accept? (y/n)
y
Added 9.0 points
Player has 129.46017612416682 points

GAME OVER
```